You're given a number N. Find the smallest number greater than N which has the same number of active bits in it (i.e. the same number of 1 in binary representation).

**Examples:**

* For N = 1 the output should be 2 (they both have 1 active bit)
* For N = 3 the output should be 5 (they both have 2 active bits)
* **[input] integer N**
  + 1 ≤ n ≤ 109.
* **[output] integer**

<https://codefights.com/challenge/RMJnEiejiiWeAfFDT>

int nextNumber(int N) {

struct Helper{

std::string toBin(int decimal) {

std::string digits = "";

while(decimal > 0){

digits.insert(digits.begin(), (decimal % 2) + '0');

decimal /= 2;

}

return digits;

}

int numUnos(std::string bin){

int unos = 0;

for(int i =0; i < bin.length(); i++){

if(bin[i] == '1'){

unos++;

}

}

return unos;

}

};

Helper h;

int unosN = h.numUnos(h.toBin(N));

int i = N + 1;

while(h.numUnos(h.toBin(i)) != unosN){

i++;

}

return i;

}